Divide and Conquer: Problem 5

- In Aceland the national sport is tennis. There is a ranking, in which each player is assigned a number of points according to their quality, that is, the best player in that country is the one with the highest number of points. Every year a couple must be selected from among all the Aceland players to play a doubles tournament at an international level. The selection procedure is a little peculiar. The score of each of the players is placed on a list, in a completely random way, without any sort of order. Once the list is made, each player can only form a pair with an adjacent player on the list, that is, who is in front of or behind him on that list. Obviously, the first player on the list can only pair with the second and the last with the penultimate, but the rest have two possible options to form the pair of doubles, corresponding to the previous and subsequent players on the list. With this restriction, the best pair of doubles possible is chosen, which is the one in which the sum of the points of its two components is greater. Design an algorithm whose main function follows the divide and conquer scheme, which decides which pair of doubles should compete in Aceland. Reason the complexity of the algorithm.

Divide and Conquer: Problem 5

```
const MaxSize= ...
type vector = array[1.. MaxSize] of integer
fun BestPair(I/O xi, xf: integer; I/O v: vector)  ret Maximum: real
var m: integer
    if ( xi == xf-1 ) then      //Basic case                          (1)
        Maximum=v(xi)+v(xf)                                           (1+1)
    else
        xm = ( xi + xf ) % 2                                          (1)
        Maximum=Max(BestPair(xi,xm,v),v(xm)+v(xm+1),
                           BestPair(xm+1,xf,v))                       2T(n/2)
    eif
    return Maximum
efun
```

$$T(n) = 1 + 2T\left(\frac{n}{2}\right)$$

Divide and Conquer: Problem 5

$$T(n) = 1 + 2T\left(\frac{n}{2}\right)$$

- Is a typical Divide and Conquer expression qith p=0, b=2 and k=2. Then:

$$\mathrm{k} > b^p$$

- Therefore:

$$\boldsymbol{T(n) \in O\left(n^{log_b k}\right) \in O(n)}$$